WEST Search History

Hide Items | Restore | Clear | Cancel

DATE: Saturday, March 20, 2004

Hide?	Set Name	Query	Hit Count
	DB=USPT; PLUR=YES; OP=ADJ		
☐	L9	l7 and l8	7
☐	L8	(709/226 or 709/203 or 709/220 or 709/223 or 718/105 or 718/104 or 718/102).ccls.	5428
☐	L7	l5 same l1	8
☐	L6	L5 and l1	43
☐	L5	l3 same L4	182
☐	L4	server near4 (select$ or assign$)	7024
☐	L3	server near2 (capabilit$ or capacit$ or characteri$)	2240
☐	L2	6205477[pn]	1
☐	L1	(server near4 request$) near8 (distribut$ or balanc$)	753

END OF SEARCH HISTORY

US006351775B1

(12) **United States Patent**
Yu

(10) **Patent No.: US 6,351,775 B1**
(45) **Date of Patent: Feb. 26, 2002**

(54) **LOADING BALANCING ACROSS SERVERS IN A COMPUTER NETWORK**

(75) Inventor: **Philip Shi-Lung Yu**, Chappaqua, NY (US)

(73) Assignee: **International Business Machines Corporation**, Armonk, NY (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **08/866,461**

(22) Filed: **May 30, 1997**

(51) **Int. Cl.**[7] **G06F 15/173**
(52) **U.S. Cl.** **709/238**; 709/239; 709/240; 709/241; 709/242; 370/237; 370/400
(58) **Field of Search** 709/242, 239, 709/240, 241, 238; 370/237, 400

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,371,852	A	12/1994	Attanasio et al.	
5,517,620	A *	5/1996	Hashimoto et al.	709/242
5,526,414	A *	6/1996	Bedard et al.	379/221
5,544,313	A *	8/1996	Shachnai et al.	709/219
5,828,847	A *	10/1998	Gehr et al.	709/239
5,864,535	A *	1/1999	Basilico	370/231
5,930,348	A *	7/1999	Regnier et al.	379/221
6,078,943	A *	6/2000	Yu	709/105
6,091,720	A *	7/2000	Bedard et al.	370/351

FOREIGN PATENT DOCUMENTS

JP 8-214063 8/1996

OTHER PUBLICATIONS

M. Colajanni et al., "Scheduling Algorithms for Distributed Web Servers", RC 20680 (91683) Jan. 6, 1997, Computer Science/Mathematics, Research Report, 29 pages.

T. Brisco, "DNS Support for Load Balancing", Apr. 1995, 6 pages, Network Working Group, Rutgers University.

Daniel M. Dias et al., "A Scalable and Highly Available Web Server", (not dated), 8 pages, IBM Research Division, T. J. Watson Research Center, Yorktown Heights, N. Y. 10598.

Eri D. Katz et al., "A scalable HTTP server: The NCSA prototype", 1994, pp. 155–164, vol. 27, Computer Networks and ISDN Systems.

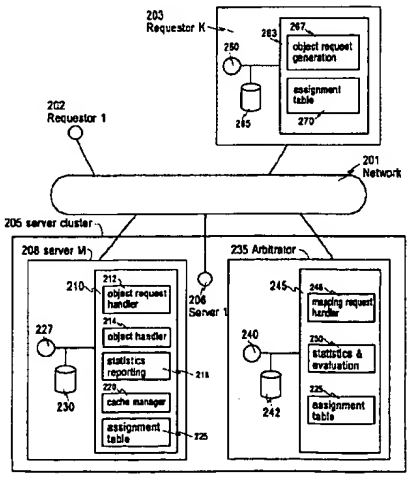
* cited by examiner

Primary Examiner—Krisna Lim
(74) *Attorney, Agent, or Firm*—F. Chau & Associates, LLP

(57) **ABSTRACT**

A dynamic routing of object requests among a collection or cluster of servers factors the caching efficiency of the servers and the load balance or just the load balance. The routing information on server location can be dynamically updated by piggybacking meta information with the request response. To improve the cache hit at the server, the server selection factors the identifier (e.g. URL) of the object requested. A partitioning method can map object identifiers into classes; and requester nodes maintain a server assignment table to map each class into a server selection. The class-to-server assignment table can change dynamically as the workload varies and also factors the server capacity. The requester node need only be informed on an "on-demand" basis on the dynamic change of the class-to-server assignment (and thus reduce communication traffic). In the Internet, the collection of servers can be either a proxy or Web server cluster and can include a DNS and/or TCP-router. The PICS protocol can be used by the server to provide the meta information on the "new" class-to-server mapping when a request is directed to a server based on an invalid or obsolete class-to-server mapping. DNS based routing for load balancing of a server cluster can also benefit. By piggybacking meta data with the returned object to reassign the requester to another server for future requests, adverse effects of the TTL on the load balance are overcome without increasing traffic.

75 Claims, 15 Drawing Sheets



First Hit Fwd Refs

Generate Collection Print

L7: Entry 1 of 8 File: USPT Feb 26, 2002

DOCUMENT-IDENTIFIER: US 6351775 B1
TITLE: Loading balancing across servers in a computer network

Abstract Text (1):
A dynamic routing of object requests among a collection or cluster of servers factors the caching efficiency of the servers and the load balance or just the load balance. The routing information on server location can be dynamically updated by piggybacking meta information with the request response. To improve the cache hit at the server, the server selection factors the identifier (e.g. URL) of the object requested. A partitioning method can map object identifiers into classes; and requester nodes maintain a server assignment table to map each class into a server selection. The class-to-server assignment table can change dynamically as the workload varies and also factors the server capacity. The requester node need only be informed on an "on-demand" basis on the dynamic change of the class-to-server assignment (and thus reduce communication traffic). In the Internet, the collection of servers can be either a proxy or Web server cluster and can include a DNS and/or TCP-router. The PICS protocol can be used by the server to provide the meta information on the "new" class-to-server mapping when a request is directed to a server based on an invalid or obsolete class-to-server mapping. DNS based routing for load balancing of a server cluster can also benefit. By piggybacking meta data with the returned object to reassign the requester to another server for future requests, adverse effects of the TTL on the load balance are overcome without increasing traffic.

Detailed Description Text (30):
In a preferred embodiment, the DNS (167) collects the number of requests issued from each requester and will generate a requester-to-server assignment table to balance the load among the servers. (For heterogeneous servers, the assigned load can be made proportional to the server's processing capacity). When a (name-to-address) mapping request arrives at the DNS (167), a server (161 . . . 163) is assigned based on the requester name (or IP address) in the assignment table. The mapping is hierarchical and multi-level, e.g., URL=>Class=>virtual server=>server. The DNS (167) can collect the load statistics and update the assignment table (225) based on a measurement interval (much) smaller than the TTL. Thus, a new assignment table can be quickly generated, to better reflect load conditions. All servers (161 . . . 163) get the up-to-date version of the assignment table (225) from the DNS (167). As before, the requesters (110 . . . 153) need not be informed of the change; they can still send requests based on the previous (name-to-address) mapping. However, if a server receives a request from a requester that is no longer assigned to that server, the server will inform the requester of the server (161 . . . 163) to which future requests should be issued. The current request will still be served and the new assignment information can be piggybacked, e.g., using PICS or a similar mechanism, with the response or returned object. When a server is overloaded, it can send an alarm signal to the DNS (167). Each time an alarm is received, the DNS (167) can recalculate the assignment table to reduce the number of requesters assigned to any overloaded servers. The requesters can also be partitioned into classes so that the assignment table can then become a class-to-server assignment.

US006324580B1

(12) **United States Patent**
Jindal et al.

(10) **Patent No.: US 6,324,580 B1**
(45) **Date of Patent: *Nov. 27, 2001**

(54) **LOAD BALANCING FOR REPLICATED SERVICES**

(75) Inventors: **Anita Jindal**; **Swee Boon Lim**, both of Cupertino; **Sanjay Radia**, Fremont; **Whei-Ling Chang**, Saratoga, all of CA (US)

(73) Assignee: **Sun Microsystems, Inc.**, Palo Alto, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **09/146,848**

(22) Filed: **Sep. 3, 1998**

(51) **Int. Cl.**[7] **G06F 15/16**; G06F 9/00
(52) **U.S. Cl.** **709/228**; 709/105
(58) **Field of Search** 709/223, 232, 709/238, 244, 105, 102, 228; 370/232; 712/27

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,371,852	12/1994	Attanasio et al.	395/200
5,459,837	10/1995	Caccavale	395/184.01
5,583,994	12/1996	Rangan	395/200
5,742,598	4/1998	Dunn et al.	370/393
5,774,660 *	6/1998	Brendel et al.	709/201

(List continued on next page.)

FOREIGN PATENT DOCUMENTS

0715257 A1	6/1996	(EP)	G06F/9/46
WO 98/26559	6/1998	(WO)	H04L/29/12

OTHER PUBLICATIONS

Colajanni, M., Yu, P. and Dias, D., Analysis of Task Assignment Policies in Scalable Distributed Web–Server Systems, IEEE Transactions on Parallel and Distributed Systems, vol. 9, No. 6, Jun. 1998, pp. 585–599.

Pending, U.S. Patent Application Serial No. 09/146,771, by Anita Jindal, et al., titled "System for Responding to a Resource Request," filed Sep. 3, 1998, with Attorney Docket No. SUN–P3317–JTF.

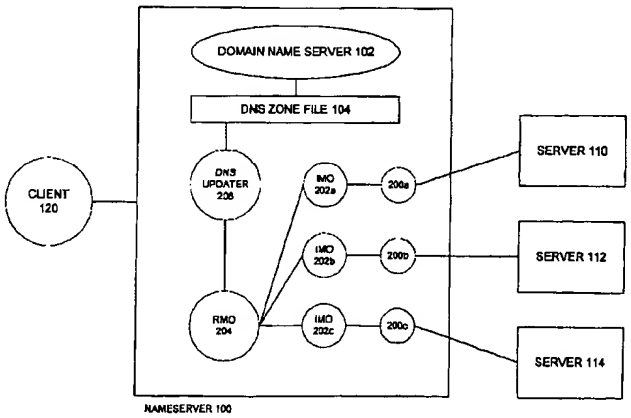
Pending U.S. Patent Application Serial No. 09/146,772, by Anita Jindal, et al., titled "Load Balancing in a Network Environment," filed Sep. 3, 1998, with Attorney Docket No. SUN–P3374–JTF.

Primary Examiner—Zarni Maung
Assistant Examiner—Jason D. Cardone
(74) *Attorney, Agent, or Firm*—Park, Vaughan & Fleming LLP

(57) **ABSTRACT**

A method is provided for load balancing requests for a replicated service or application among a plurality of servers operating instances of the replicated service or application. A policy is selected for choosing a preferred server from the plurality of servers according to one or more specified status or operational characteristics of the servers, such as the least-loaded or closest server. The policy is encapsulated within multiple levels of objects or modules that are distributed among the servers offering the replicated service and a central server that receives requests for the service. Status objects gather or retrieve information concerning the specified status or operational characteristic(s) of each of the plurality of servers. An individual server monitor object operates for each instance of the replicated service to invoke one or more status objects and receive the necessary information. A central replicated monitor object receives the information from each individual server monitor object. The information from the servers is analyzed to select the server having the optimal status or operational characteristic(s). An update object updates the central server, such as a domain name server, to indicate the preferred server. Requests for the replicated service are then directed to the preferred server until a different preferred server is identified.

30 Claims, 5 Drawing Sheets



First Hit Fwd Refs

Generate Collection Print

L7: Entry 2 of 8 File: USPT Nov 27, 2001

DOCUMENT-IDENTIFIER: US 6324580 B1
TITLE: Load balancing for replicated services

Abstract Text (1):
A method is provided for load balancing requests for a replicated service or application among a plurality of servers operating instances of the replicated service or application. A policy is selected for choosing a preferred server from the plurality of servers according to one or more specified status or operational characteristics of the servers, such as the least-loaded or closest server. The policy is encapsulated within multiple levels of objects or modules that are distributed among the servers offering the replicated service and a central server that receives requests for the service. Status objects gather or retrieve information concerning the specified status or operational characteristic(s) of each of the plurality of servers. An individual server monitor object operates for each instance of the replicated service to invoke one or more status objects and receive the necessary information. A central replicated monitor object receives the information from each individual server monitor object. The information from the servers is analyzed to select the server having the optimal status or operational characteristic(s). An update object updates the central server, such as a domain name server, to indicate the preferred server. Requests for the replicated service are then directed to the preferred server until a different preferred server is identified.

US006078943A

United States Patent [19]

Yu

[11] Patent Number: 6,078,943

[45] Date of Patent: Jun. 20, 2000

[54] **METHOD AND APPARATUS FOR DYNAMIC INTERVAL-BASED LOAD BALANCING**

[75] Inventor: **Philip Shi-Lung Yu**, Chappaqua, N.Y.

[73] Assignee: **International Business Machines Corporation**, Armonk, N.Y.

[21] Appl. No.: **08/798,385**

[22] Filed: **Feb. 7, 1997**

[51] **Int. Cl.**[7] **G06F 9/00**

[52] **U.S. Cl.** **709/105**; 709/102

[58] **Field of Search** 709/102, 103, 709/105, 100

[56] **References Cited**

U.S. PATENT DOCUMENTS

5,287,508	2/1994	Hejna, Jr. et al.	709/102
5,371,852	12/1994	Attanasio et al.	709/245
5,774,668	6/1998	Choquier et al.	709/223
5,815,664	9/1998	Asano	709/227

FOREIGN PATENT DOCUMENTS

WO9729424	8/1997	WIPO	G06F 9/46

OTHER PUBLICATIONS

Brisco, T., "DNS Support for Load Balancing", RFC 1974, Rutgers University, Apr. 1995.

Dias, D. M. et al., "A Scalable and Highly Available Web Server", Proc. 41st IEEE Computer Society Intl. Conf. (COMPCON) 1996, Technologies for the Information Superhighway, pp. 85–92, Feb. 1996.

Katz, E. et al. "A Scalable HTTP Server: The NCSA Prototype", Computer Networks and ISDN Systems, vol. 27, 1994, pp. 68–74.

Colajanni, M. et al. "Scheduling Algorithms for Distributed Web Servers", IBM Research Report, RC 20680, Jan. 1997.

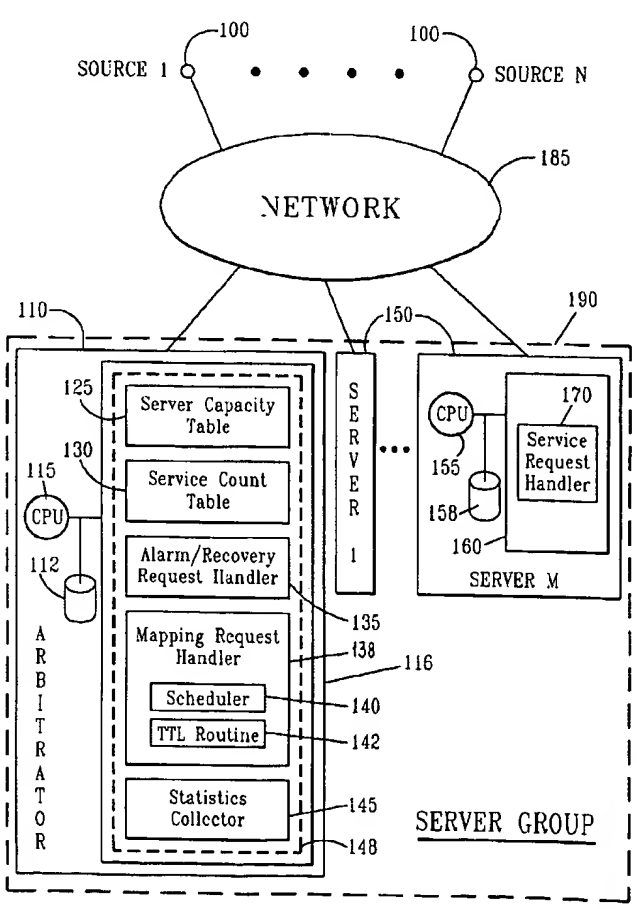
Primary Examiner—Majid A. Banankhah

Attorney, Agent, or Firm—Kevin M. Jordan

[57] **ABSTRACT**

A method for use in geographically distributed or clustered system wherein an arbiter assigns clients to servers. The arbiter also dynamically assigns a valid time interval to each mapping request based on network load and/or capacity parameters such as the client request rate and/or the server capacity. Alternative means for dynamically setting the valid interval in conjunction with a scheduling process, which can be either deterministic or probabilistic, are also devised.

20 Claims, 9 Drawing Sheets



L7: Entry 5 of 8 File: USPT Jun 20, 2000

DOCUMENT-IDENTIFIER: US 6078943 A
TITLE: Method and apparatus for dynamic interval-based load balancing

Brief Summary Text (19):
A method having features of the present invention can be embodied in a distributed or clustered network of servers wherein clients are divided into groups which periodically send mapping requests to an arbitrator for mapping and balancing service requests among multiple replicated servers which can service the request. An example of a computerized method according to the present invention for mapping servers to service requests includes the steps of: mapping a first mapping request from a first group to a first server according to a scheduling process; dynamically computing a valid interval for said mapping request to the first server as a function of one of a first group request load and a first server capacity; and communicating the server selection and the valid interval to the first group for caching such that subsequent requests from the first group are routed to the first server during the valid interval.